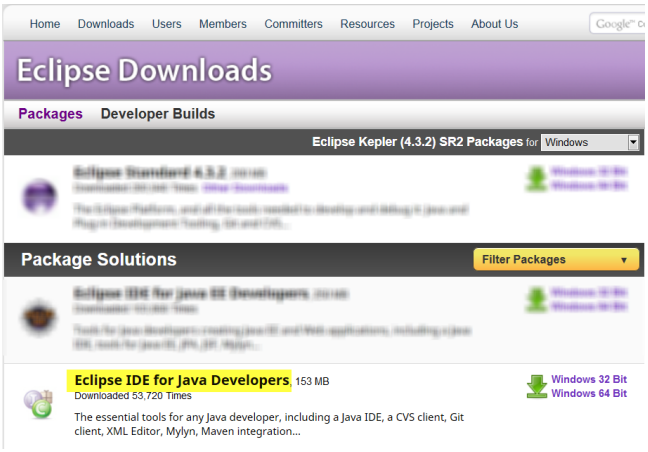
TOBB ETU
HADOOP - IBM BigInsights
Örnek Uygulama

İrfan Bahadır KATİPOĞLU*

6 Mart 2014

1 Çalışma Ortamının Edinilmesi

Eclipse çalışma ortamının *"Eclipse IDE for Java EE Developers"* sürümünü indiriniz.[i]



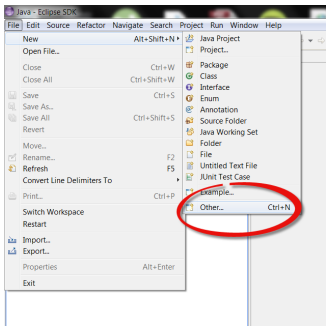
Şekil 1: Eclipse Download Page

*ibkatipoglu@etu.edu.tr — bahadir@bahadir.me
[i]http://www.eclipse.org/downloads/

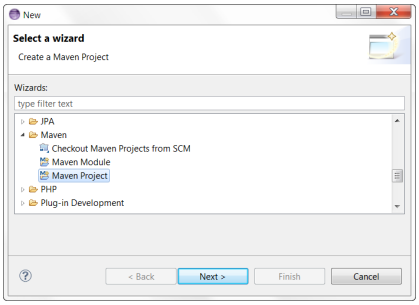
2 Projenin Geliştirilmesi

2.1 Maven Projesinin Hazırlanması

File → New → Other (Ctrl+N) ile gelen penceredeki Maven bölümünden *Maven Project*'i seçiniz.



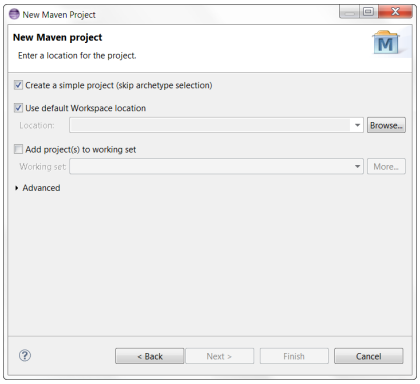
(a) Yeni Proje



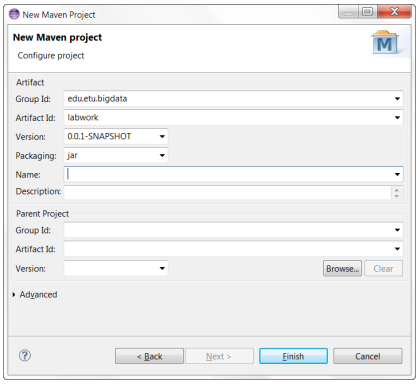
(b) Maven Project

Şekil 2: Maven Projesi

Sihirbazın sonraki ekranlarını şekildeki gibi tamamlayınız.



(a) Proje Konumu ve İçeriği



(b) Artifact Bilgileri

Şekil 3: Maven Proje Sihirbazı

Sihirbaz tamamlandıktan sonra proje iskeleti oluşacaktır. Oluşan projedeki pom.xml dosyasını açarak Kod-1[i] ile aynı olacak şekilde düzenleyiniz.

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" ↙
      ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 ↙
      ↳ http://maven.apache.org/xsd/maven-4.0.0.xsd">
3 <modelVersion>4.0.0</modelVersion>
4 <groupId>edu.etu.bigdata</groupId>
5 <artifactId>labwork</artifactId>
6 <version>0.0.1-SNAPSHOT</version>
7
8 <properties>
```

[i]Bu kodu *https://gist.github.com/bahadrix/9366787* adresinde de bulabilirsiniz.

```
<finalName>hadoop-labwork</finalName>
<mainClass>WordCount</mainClass>
</properties>

<dependencies>
    <dependency> <!-- Galaxy -->
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-client</artifactId>
    <version>2.2.0</version>
    </dependency>

    <dependency> <!-- Unit Testing ability for map reduce jobs -->
    <groupId>org.apache.mrunit</groupId>
    <artifactId>mrunit</artifactId>
    <version>0.9.0-incubating</version>
    <classifier>hadoop1</classifier>
    </dependency>

    <dependency> <!-- Apache's Common Utilities -->
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-io</artifactId>
    <version>1.3.2</version>
    </dependency>
</dependencies>

<build>
<plugins>

<plugin><!-- needed for debugging in IntelliJ IDEA -->
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-surefire-plugin</artifactId>
<version>2.14</version>
<configuration>
    <forkMode>never</forkMode>
</configuration>
</plugin>

<plugin><!-- using shade plugin for uber jar -->
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-shade-plugin</artifactId>
<version>2.2</version>
<executions>
    <execution>
        <id>main</id>
        <phase>package</phase>
        <goals>
            <goal>shade</goal>
        </goals>
        <configuration>
            <finalName>${finalName}</finalName>
            <transformers>
                <transformer
                    implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
                    <mainClass>${mainClass}</mainClass>
                </transformer>
            </transformers>
            <!-- Exclude with dependencies -->
            <minimizeJar>true</minimizeJar>
            <artifactSet>
                <excludes>
                    <!-- Some libs is already included in running ↙
                        ↳ machine's classpath,
                    we exclude them -->
                    <exclude>org.apache.hadoop</exclude>
                    <exclude>junit</exclude>
                    <exclude>org.apache.mrunit</exclude>
                    <exclude>log4j</exclude>
                    <exclude>org.xerial.snappy</exclude>
                    <exclude>org.mockito</exclude>
                    <exclude>org.codehaus.jackson</exclude>
                    <exclude>org.objenesis</exclude>
                    <exclude>org.apache.avro</exclude>
                    <exclude>com.google.protobuf</exclude>
                    <exclude>com.google.common</exclude>
```

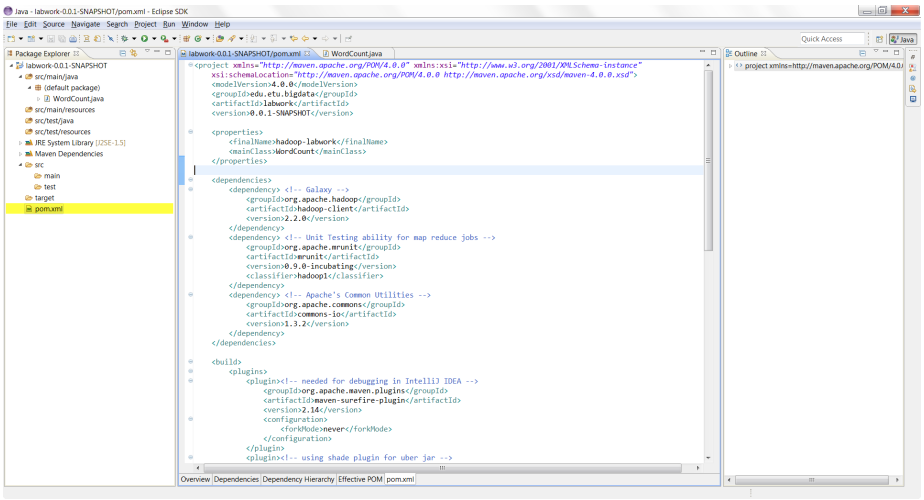
```
                        <exclude>org.slf4j</exclude>
                    </excludes>
                </artifactSet>

            </configuration>
        </execution>
</executions>
</plugin>

</plugins>
</build>

</project>
```

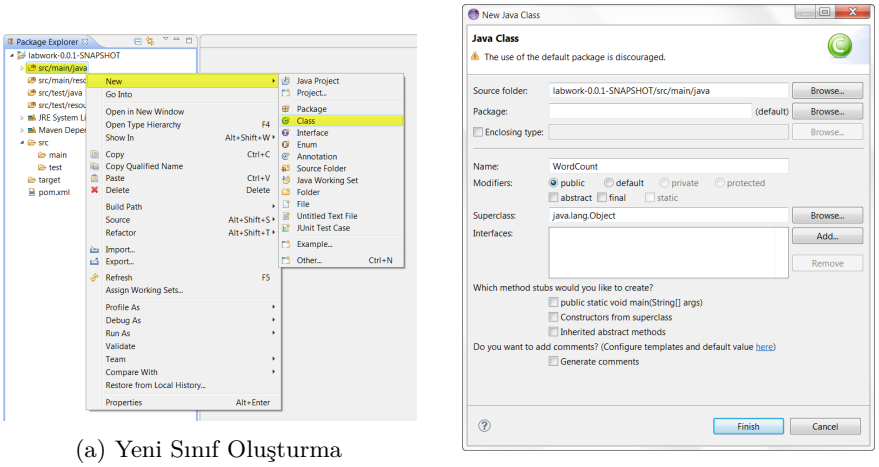
Kod 1: POM Dosyası



Şekil 4: Maven Proje İskeleti

2.1.1 Sınıfların Kodlanması

src/main/java dizinine sağ tıklayıp New→Class seçeneği ile şekildeki gibi bir *WordCount* sınıfı oluşturunuz.



(a) Yeni Sınıf Oluşturma

(b) Sınıf Özellikleri

Şekil 5: WordCount Sınıfının Oluşturulması

Bu sınıfı açarak Kod-2[i] ile aynı olacak şekilde düzenleyiniz.

```
/**
 * ETU Hadoop - BigInsights
 * Sample WordCount Class
 * @author Bahadir Kaitpoglu
 */

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

import java.io.IOException;
import java.util.Iterator;
import java.util.StringTokenizer;

public class WordCount extends Configured implements Tool {

    /**
     * Mapper class
     *
     */
    static class Mappa extends MapReduceBase implements
            Mapper<LongWritable, Text, Text, IntWritable> {

        private Text word = new Text();
        private final static IntWritable ONE = new IntWritable(1);

        public void map(LongWritable key, Text value,
                OutputCollector<Text, IntWritable> collector, Reporter arg3)
                throws IOException {

```

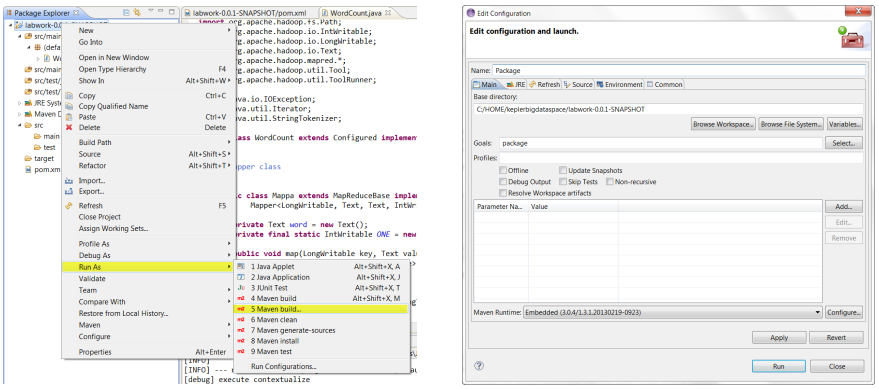
[i]Kodu *https://gist.github.com/bahadrix/9370719* adresinden de indirebilirsiniz.

```
37              String line = value.toString();
38              StringTokenizer tokenizer = new StringTokenizer(line);
39              while (tokenizer.hasMoreTokens()) {
40                  word.set(tokenizer.nextToken());
41                  collector.collect(word, ONE);
42              }
43
44          }
45
46      }
47
48      /**
49       * Reducer Class
50       *
51       */
52      static class Reducca extends MapReduceBase implements
53              Reducer<Text, IntWritable, Text, IntWritable> {
54
55          public void reduce(Text key, Iterator<IntWritable> values,
56                  OutputCollector<Text, IntWritable> collector, Reporter arg3)
57                  throws IOException {
58              int sum = 0;
59              while (values.hasNext()) {
60                  sum += values.next().get();
61              }
62              collector.collect(key, new IntWritable(sum));
63
64          }
65
66      }
67
68      /**
69       * Entry point
70       *
71       * @param args
72       * @throws Exception
73       */
74      public static void main(String[] args) throws Exception {
75          System.exit(ToolRunner.run(new Configuration(), new WordCount(), ↙
                ↳ args));
76      }
77
78      /**
79       * Run the job
80       */
81      public int run(String[] args) throws Exception {
82
83          Configuration conf = getConf();
84          JobConf job = new JobConf(conf, WordCount.class);
85
86          Path pIn = new Path(args[0]);
87          Path pOut = new Path(args[1]);
88          FileInputFormat.setInputPaths(job, pIn);
89          FileOutputFormat.setOutputPath(job, pOut);
90
91          job.setJobName("WordCount");
92          job.setMapperClass(Mappa.class);
93          job.setReducerClass(Reducca.class);
94
95          job.setOutputKeyClass(Text.class);
96          job.setOutputValueClass(IntWritable.class);
97
98          JobClient.runJob(job);
99
100         return 0;
101     }
102
103 }
```

Kod 2: WordCount Sınıfı

3 Derleme

Projenin ana dizinine sağ tıklayıp *Run as→Maven Build...* seçeneğini seçiniz. Gelen ekrandaki *goals* ve *package* alanlarını şekildeki gibi doldurup Run butonuna tıklayınız[i].

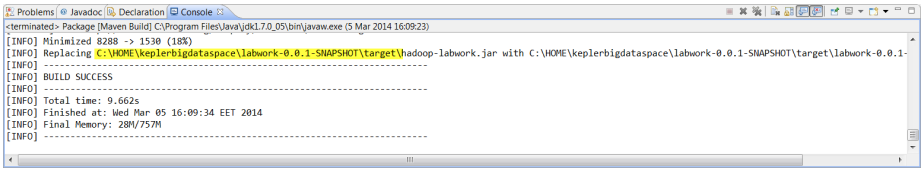


(a) Maven Build...

(b) Package

Şekil 6: Maven Package Konfigurasyonunun Oluşturulması

Maven paketlemeyi başardığında aşağıdaki gibi bir konsol çıktısı vermeli. Şekilde işaretlenmiş olan adres Hadoop ile kullanılmak üzere oluşturulmuş olan JAR dosyasının bilgisayardaki konumunu gösteriyor. Bu konumu kopyalıyoruz.



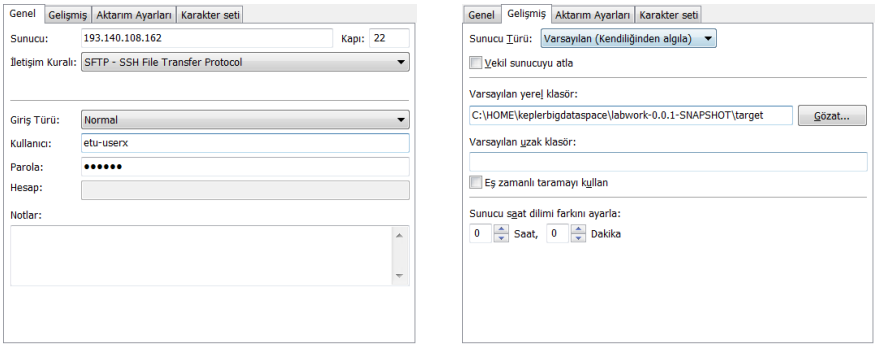
Şekil 7: Maven Paketleme İşlemi Konsol Çıktısı

4 Yükleme

4.1 SFTP İşlemleri

FileZilla adlı programı *https://filezilla-project.org/download.php?show_all=1* adresinden indirerek kurunuz. Programı çalıştırıp Ctr + S tuşlarına basarak gelen SiteManager ekranından New Site butonuna tıklayarak yapılandırmasını aşağıdaki şekilde gösterildiği gibi yapınız. Yerel dizin olarak jar dosyasının bulunduğu dizinin adresini yazınız.

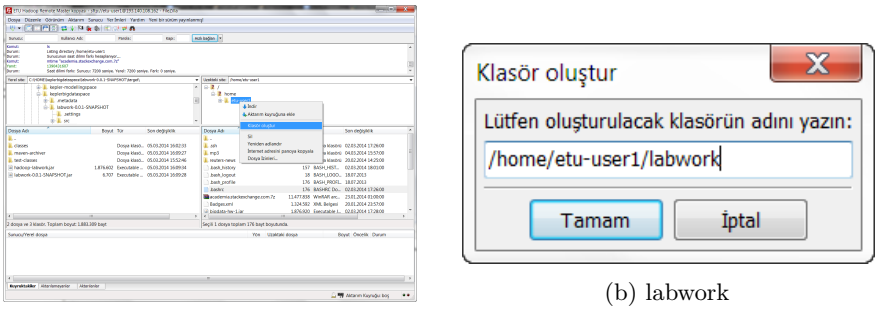
[i]Daha sonradan projeyi tekrar paketlemek için Run→Run History→Package menüsünü kullanabilirsiniz.



(a) Sunucu ayarları (b) Yerel dizin ayarı

Şekil 8: FileZilla SFTP erişiminin oluşturulması

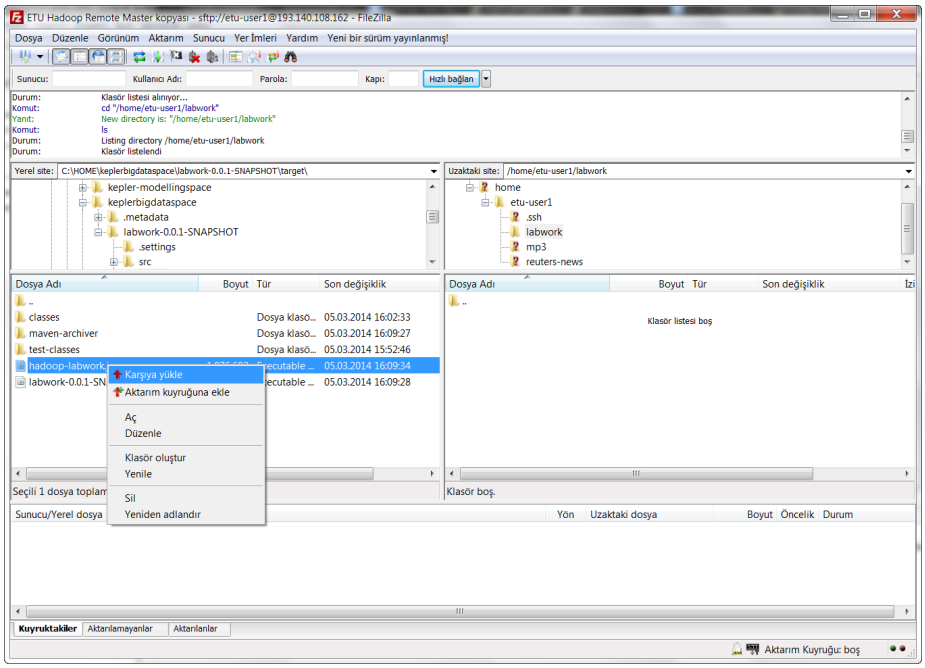
Bağlan butonuna basarak MasterNode'a bağlanınız. Sol taraf bağlanan bilgisayarın yerel dizinini, sağ taraf sunucu tarafını göstermektedir.
Bağlantı kurulduktan sonra aşağıda gösterildiği gibi sunucudaki dizininize *labwork* adlı bir altdizin ekleyiniz.



(a) Kullanıcı dizini altında yeni klasör (b) labwork

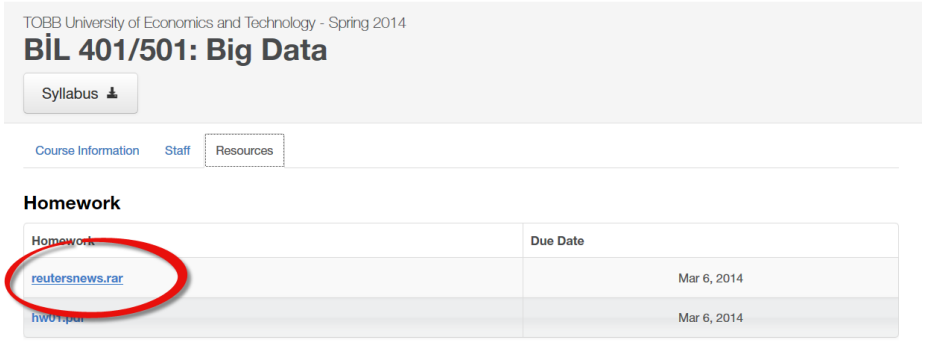
Şekil 9: Çalışma dizininin oluşturulması

Daha sonra oluşturmuş olduğunuz jar dosyasını bu yeni dizine kopyalayınız.



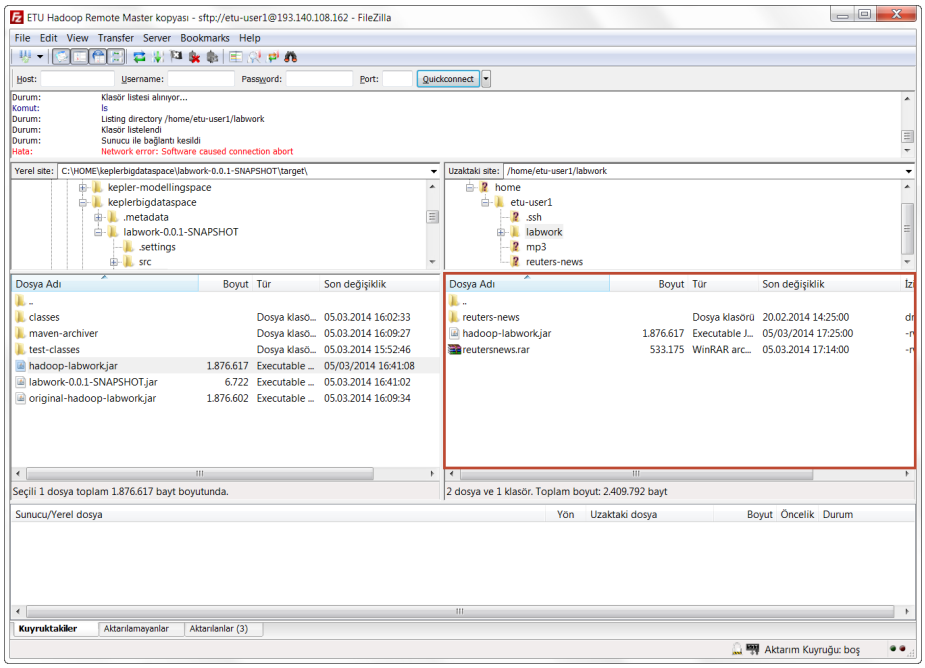
Şekil 10: JAR Dosyasının Transferi

Bu işlemden sonra ödevde verilmiş olan reuters-news.rar dosyasını indiriniz.



Şekil 11: Piazza - ReutersNews

İndirdiğiniz dosyayı Şekil 12 ile gösterilen alana sürükleyip bırakarak sunucudaki aynı dizine transferini sağlayınız.



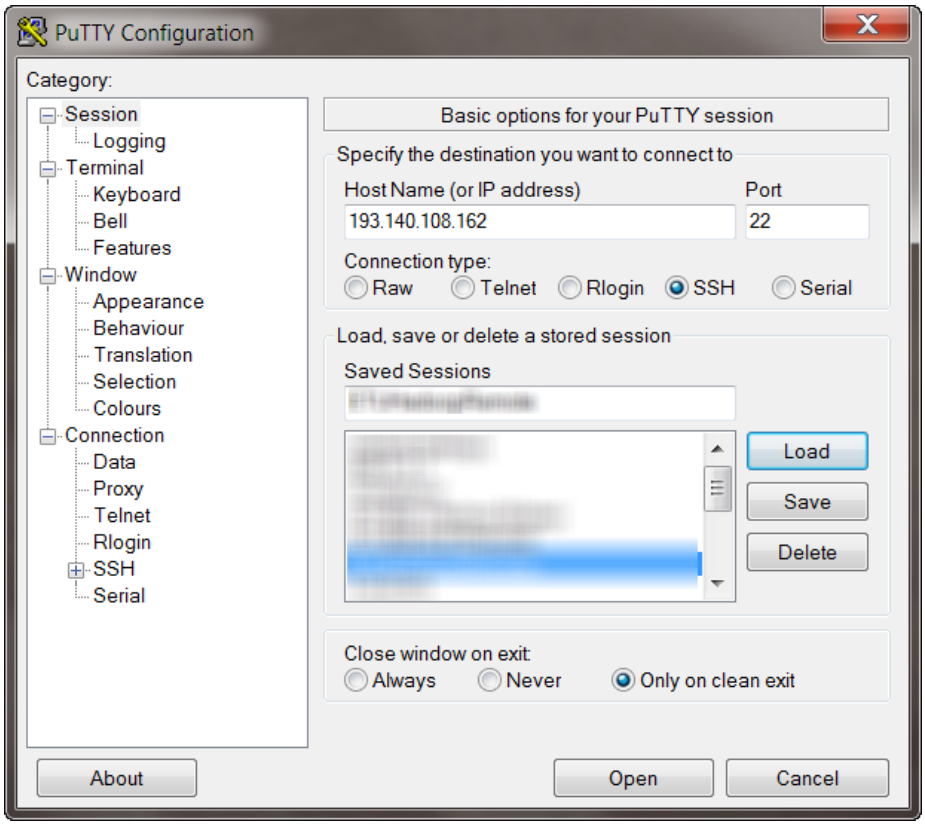
Şekil 12: Veri Setinin Transferi

5 Uygulamanın Hadoop İle Yürütülmesi

5.1 SSH Bağlantısının Sağlanması

Windows kullanıcıları Putty ile şekilde gösterildiği gibi sunucuya bağlanabilirler. Linux tabanlı sistemlerde aşağıdaki gibi sunuya SSH bağlantısı kurulabilir.

```
ssh 193.140.108.162
```



Şekil 13: Putty yapılandırma ekranı

5.2 SSH İşlemleri

```
$ cd labwork

# Check our uploaded files
$ ls
hadoop-labwork.jar reutersnews.rar

# Extract dataset
$ unrar x reutersnews.rar

# Create folders for dataset
$ hadoop dfs -mkdir labwork/input

# Put files to the HDFS.
# time is optional for measuring time
$ time hadoop dfs -put reuters-news/*.txt labwork/input
real    0m6.506s
user    0m5.870s
sys     0m0.702s

# Run!
$ hadoop jar hadoop-labwork.jar labwork/input labwork/output

# Check the output
$ hadoop dfs -ls labwork/output
Found 3 items
-rw-r--r--   3 ... /user/etu-user1/labwork/output/_SUCCESS
drwxr-xr-x   - ... /user/etu-user1/labwork/output/_logs
-rw-r--r--   3 ... /user/etu-user1/labwork/output/part-00000

# Print out the result
$ hadoop dfs -cat labwork/output/*0
(...)
zero    4
zero,   1
zero-coupon     3
zestril,        2
zinc    1
zone    4
zone,   1
zones,  1
```

Kod 3: Hadoop Görevinin İşletilmesi

Çalışmakta olan ve geçmiş Job'lara ait durumu BigInsights Web Console'dan Application Status bölümünden izleyebilirsiniz.